

TO:           Distribution  
FROM:         Joan Archer  
DATE:         30 May 75  
RE:           Multics Change Requests

Enclosed are copies of Multics Change Requests which were approved from 16 May 75 to 31 May 75.

---

Multics Project internal working documentation. Not to be reproduced or distributed outside the Multics Project.

TITLE: Fix ioam to check size of IOAT		STATUS	DATE	
AUTHOR: M. J. Grady		Written	05/20/75	
-Coded in: <input type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Status	
	Lib. Maint. Tools		A 05/27/75	
	Sys. Anal. Tools		Expires	
	Sys. Prog. Tools		11/27/75	
	355		DOCUMENTATION CHANGES	
	BOS		Document	Specify One or More
	Salvager		MPM (Vol, Sect.)	
	Ring Zero		PLMS (AN #)	
	Ring One		MOSN (Sect.)	
	SysDaemon/Admin.		MPAM (Sect.)	
Runtime		MSAM (Sect.)		
User Cmmd/Subr.		Info Segs		
Objections/Comments:		Other (Name)		
		None (Reason)		

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

SUMMARY:

Fix the i/o assignment manager, ioam, to check the size of the IOAT before adding a new entry. If the IOAT is too small, type a message to the operator and return error\_table\_\$ioat\_err to user if possible.

REASONS:

Currently the ioam does not check, and just adds the new entry. If the IOAT is too small, an out\_of\_bounds fault occurs and the process is terminated. The Initializer then attempts to get the device back and delete the IOAT entry, which results in another out\_of\_bounds fault, however this terminates the Initializer, and thus the system.

IMPLICATIONS:

None.

Multics Change Request

TITLE: Fix bug in tty_lines		STATUS	DATE	
AUTHOR: D. R. Vinograd		Written	05/12/75	
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input checked="" type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Status A 05/21/75	
		<input type="checkbox"/> Lib. Maint. Tools	Expires 11/21/75	
		<input checked="" type="checkbox"/> Sys. Anal. Tools	DOCUMENTATION CHANGES	
		<input type="checkbox"/> Sys. Prog. Tools	Document	Specify One or More
		<input type="checkbox"/> 355		
	<input type="checkbox"/> BOS	MPM (Vol, Sect.)		
	<input type="checkbox"/> Salvager	PLMS (AN #)		
	<input type="checkbox"/> Ring Zero	MOSN (Sect.)		
	<input type="checkbox"/> Ring One	MPAM (Sect.)		
	<input type="checkbox"/> SysDaemon/Admin.	MSAM (Sect.)		
	<input type="checkbox"/> Runtime			
	<input type="checkbox"/> User Cmnd/Subr.			
Objections/Comments:		Info Segs		
		Other (Name)		
		None (Reason)		

Use these headings: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (Optional)

SUMMARY:

If the command tty\_lines is executed without arguments it will randomly go into a loop.

REASONS:

This occurs because of a transfer into the middle of a "do" loop.

IMPLICATIONS:

None

DETAILED PROPOSAL:

It will be fixed.

Multics Change Request

TITLE: Fix table allocation bug in tc_init		STATUS	DATE
AUTHOR: D. R. Vinograd		Written	05/12/75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input checked="" type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Status
	<input type="checkbox"/>	Lib. Maint. Tools	A 05/21/75
	<input type="checkbox"/>	Sys. Anal. Tools	Expires
	<input type="checkbox"/>	Sys. Prog. Tools	11/21/75
<input type="checkbox"/>	355	DOCUMENTATION CHANGES	
<input type="checkbox"/>	BOS	Document	Specify One or More
<input type="checkbox"/>	Salvager	MPM (Vol, Sect.)	
<input checked="" type="checkbox"/>	Ring Zero	PLMS (AN #)	
<input type="checkbox"/>	Ring One	MOSN (Sect.)	
<input type="checkbox"/>	SysDaemon/Admin.	MPAM (Sect.)	
<input type="checkbox"/>	Runtime	MSAM (Sect.)	
<input type="checkbox"/>	User Cmmd/Subr.	Info Segs	
Objections/Comments:		Other (Name)	
		None (Reason)	

Use these headings: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (Optional)

SUMMARY:

tc\_init does not check whether the APT, ITT, or DST will fit within the specified length of tc\_data.

REASONS:

If either the APT or ITT will not fit, the system will crash, because of a bounds fault, without a reasonable message. If there is no room for the DST, funny messages will occur during device attachment and processes will terminate.

DETAILED PROPOSAL:

Put in a check for required space and crash the system with a message as to how much storage is needed if size field of TCD config card is not correct.

Multics Change Request

TITLE: Modifications to network_exec_		STATUS	DATE		
AUTHOR: Kenneth T. Pogram		Written	05/12/75		
<input checked="" type="checkbox"/> Coded in PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL <input type="checkbox"/> Planned for System MR <input type="checkbox"/> Fixes Bug Number(s) _____ <input type="checkbox"/> Documented in MTB _____ <input checked="" type="checkbox"/> <del>User</del> /Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no <input type="checkbox"/> Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse <input type="checkbox"/> Replaces MCR _____		Status	A 05/21/75		
		Expires	11/21/75		
		DOCUMENTATION CHANGES		Document	Specify One or More
		Category (Check One)		MPM (Vol, Sect.)	
			PLMS (AN #) 64		
			MOSN (Sect.)		
			MPAM (Sect.)		
			MSAM (Sect.)		
Objections/Comments:		Info Segs			
		Other (Name)			
		None (Reason)			

Use these headings: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (Optional)

SUMMARY: network\_exec\_, the process overseer for the Network Daemon, is modified as follows:

1. Reporting of resource consumption is modified. Call to hcs\_\$get\_usage\_values is replaced with call to hcs\_\$get\_process\_usage so that virtual CPU time and memory units can be reported.
2. All-upper-case messages printed on the BOS console via calls to phcs\_\$ring\_0\_message (called by net\_log\_) are replaced with mixed-case messages.
3. The status code returned by netp\_\$ncp\_daemon\_wakeup is checked to see if it is negative; such a code returned is I/O status from the IMP DIM and is reported as such, rather than being considered an error.
4. A new message is printed on the BOS console when operator action is required.

- REASONS:
1. Resource consumption was not being reported properly.
  2. With the new syserr\_mechanisms, all-upper-case messages result in lots of annoying backslashes on the BOS console.
  3. If an I/O status code is treated as an error, the Network Daemon will not automatically re-initialize the NCP in the event of an IMP or ABSI failure.
  4. The Network Daemon prints messages on its own terminal as it takes various actions in an attempt to restore normal Network service. At a busy installation running the message coordinator, with lots of messages being printed, a message on the Network Daemon's I/O stream indicating that operator action is required tends to be overlooked. The specific message, "Operator action required" printed by the Network Daemon on the BOS console will alert the operator in the (relatively rare) event that he must take some action.

Ver. 3	MULTICS CHANGE REQUEST	MCR_1180
741022		
TITLE: ascii_to_ebcdic_ and ebcdic_to_ascii_ conversion routines		STATUS   DATE
AUTHOR: Ross E. Klinger		Written   05/05/75
		Status   H 05/21/75
		Expires   11/05/75
Planned for System: MR 3.0		
Fixes Bug Number(s): not applicable		CATEGORY (check one)
Documented in MTB: MTB-90		( ) Lib. Maint. Tools
Incompatible Change: no		( ) Sys. Anal. Tools
User/Operations-visible Interface Change: no		( ) Sys. Prog. Tools
Coded in: ( ) PL/I ( <input checked="" type="checkbox"/> ) ALM ( ) other-see below		( ) 355
Performance: ( ) better ( <input checked="" type="checkbox"/> ) same ( ) worse		( ) BOS
		( ) Salvager
DOCUMENTATION CHANGES (specify one or more)		( ) Ring Zero
MPM (vol,sect) II,10 MPAM (sect)		( ) Ring One
MOSN (sect) MSAM (sect)		( ) SysDaemon/Admin
PLMs (AN#) AN57		( ) Runtime
Info Segs ascii_to_ebcdic_.info		( <input checked="" type="checkbox"/> ) User Command/Subr
ebcdic_to_ascii_.info		
OBJECTIONS/COMMENTS:		

Headings are: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (optional)

SUMMARY: make the ascii\_to\_ebcdic\_ and ebcdic\_to\_ascii\_ conversion routines available to the user. These procedures are currently internal interfaces in the tape\_ansi\_ and tape\_ibm\_ I/O modules.

REASON: A mechanism should be provided for performing ASCII-EBCDIC/EBCDIC-ASCII conversion in a uniform manner.

IMPLICATIONS: a system standard will be created.

DETAILED PROPOSAL: see the attached draft MPM documentation. Note that the proposed mapping differs slightly from that published in MTB-90, for left-bracket ([), right-bracket (]), and broken-bar (!). The final mapping has been developed considering the following:

- 1) the OS ASCII-EBCDIC translate SVC;
- 2) OS TCAM mappings for ASCII terminals;
- 3) Type TN and Princeton Preferred Print Train encodings;
- 4) proposed and implemented network mappings.

**Name:** ascii\_to\_ebcdic\_

This subroutine performs isomorphic (one-to-one reversible) conversion from ASCII to EBCDIC. The input data is a string of valid ASCII characters. A valid ASCII character is defined as a 9 bit byte with an octal value in the range  $0 \leq \text{octal\_value} \leq 177$ .

**Entry:** ascii\_to\_ebcdic\_

This entry accepts an ASCII character string, and generates an EBCDIC character string of equal length.

#### Usage

```
dcl ascii_to_ebcdic_ entry (char (*), char (*));
```

```
call ascii_to_ebcdic_ (ascii_in, ebcdic_out);
```

- 1) `ascii_in` are the ASCII characters to be converted. (Input)
- 2) `ebcdic_out` is the EBCDIC equivalent of the input string. (Output)

**Entry:** ascii\_to\_ebcdic\_\$table

This entry defines the 128 character translation table used to perform conversion from ASCII to EBCDIC. The mappings implemented by `ascii_to_ebcdic_` and `ebcdic_to_ascii_` are isomorphic; i.e., every valid character has a unique mapping, and mappings are reversible. (See the MPM write-up of the `ebcdic_to_ascii_` subroutine.) The result of an attempt to convert a character which is not in the ASCII character set is undefined.

#### Usage

```
dcl ascii_to_ebcdic_$table char (128) external static;
```

ISOMORPHIC ASCII/EBCDIC CONVERSION TABLE

ASCII		EBCDIC	
GRAPHIC	OCTAL	HEXADECIMAL	GRAPHIC
NUL	000	00	NUL
SOH	001	01	SOH
STX	002	02	STX
ETX	003	03	ETX
EOT	004	37	EOT
ENQ	005	20	ENQ
ACK	006	2E	ACK
BEL	007	2F	BEL
BS	010	16	BS
HT	011	05	HT
LF	012	25	NL
VT	013	08	VT
FF	014	0C	NP
CR	015	0D	CR
SO	016	0E	SO
SI	017	0F	SI
DLE	020	10	DLE
DC1	021	11	DC1
DC2	022	12	DC2
DC3	023	13	TM
DC4	024	3C	DC4
NAK	025	3D	NAK
SYN	026	32	SYN
ETB	027	26	ETB
CAN	030	18	CAN
EM	031	19	EM
SUB	032	3F	SUB
ESC	033	27	ESC
FS	034	1C	IFS
GS	035	1D	IGS
RS	036	1E	IRS
US	037	1F	IUS



GRAPHIC	OCTAL	HEXADECIMAL	GRAPHIC
space	040	40	space
!	041	5A	!
"	042	7F	"
#	043	7B	#
\$	044	5B	\$
%	045	6C	%
&	046	50	&
.	047	7D	.
(	050	4D	(
)	051	5D	)
*	052	5C	*
+	053	4E	+
,	054	6B	,
-	055	60	-
.	056	4B	.
/	057	61	/
0	060	F0	0
1	061	F1	1
2	062	F2	2
3	063	F3	3
4	064	F4	4
5	065	F5	5
6	066	F6	6
7	067	F7	7
8	070	F8	8
9	071	F9	9
:	072	7A	:
;	073	5E	;
<	074	4C	<
=	075	7E	=
>	076	6E	>
?	077	6F	?

GRAPHIC	OCTAL	HEXADECIMAL	GRAPHIC
	100	7C	
A	101	C1	A
B	102	C2	B
C	103	C3	C
D	104	C4	D
E	105	C5	E
F	106	C6	F
G	107	C7	G
H	110	C8	H
I	111	C9	I
J	112	D1	J
K	113	D2	K
L	114	D3	L
M	115	D4	M
N	116	D5	N
O	117	D6	O
P	120	D7	P
Q	121	D8	Q
R	122	D9	R
S	123	E2	S
T	124	E3	T
U	125	E4	U
V	126	E5	V
W	127	E6	W
X	130	E7	X
Y	131	E8	Y
Z	132	E9	Z
[	133	AD	[ (see <u>Note</u> )
\	134	ED	\
]	135	BD	] (see <u>Note</u> )
^	136	5F	logical NOT
_	137	6D	_

GRAPHIC	OCTAL	HEXADECIMAL	GRAPHIC
~	140	79	~
a	141	81	a
b	142	82	b
c	143	83	c
d	144	84	d
e	145	85	e
f	146	86	f
g	147	87	g
h	150	88	h
i	151	89	i
j	152	91	j
k	153	92	k
l	154	93	l
m	155	94	m
n	156	95	n
o	157	96	o
p	160	97	p
q	161	98	q
r	162	99	r
s	163	A2	s
t	164	A3	t
u	165	A4	u
v	166	A5	v
w	167	A6	w
x	170	A7	x
y	171	A8	y
z	172	A9	z
[	173	C0	[
	174	4F	solid bar
]	175	D0	]
~	176	A1	~
DEL	177	07	DEL

```
|-----|
|  ascii_to_ebcdic_  |
|-----|
```

MULTICS PROGRAMMERS' SUPPLEMENT

Page 6

Note

These graphics ("[" and "]") do not appear in (or map into any graphics which appear in) the standard EBCDIC character set. They have been assigned to otherwise "illegal" EBCDIC code values, in conformance with the bit patterns used by the TN text printing train.

Calling `ascii_to_ebcdic_` is as efficient as using the PL/I `translate` builtin, as conversion is performed by a single MVT instruction, and the procedure runs in the stack frame of its caller.

**Name:** ebcDic\_to\_ascii\_

This subroutine performs isomorphic (one-to-one reversible) conversion from EBCDIC to ASCII. The input data is a string of valid EBCDIC characters. A valid EBCDIC character is defined as a 9 bit byte with a hexadecimal value in the range  $00 \leq \text{hex\_value} \leq \text{FF}$  (octal value in the range  $000 \leq \text{oct\_value} \leq 377$ ).

**Entry:** ebcDic\_to\_ascii\_

This entry accepts an EBCDIC character string, and generates an ASCII character string of equal length.

**Usage**

```
dcl ebcDic_to_ascii_ entry (char (*), char (*));
```

```
call ebcDic_to_ascii_ (ebcDic_in, ascii_out);
```

- 1) ebcDic\_in are the EBCDIC characters to be converted.  
(Input)
- 2) ascii\_out is the ASCII equivalent of the input string.  
(Output)

**Entry:** ebcDic\_to\_ascii\_\$table

This entry defines the 256 character translation table used to perform conversion from EBCDIC to ASCII. Of the 256 valid EBCDIC characters, only 128 have ASCII equivalents. These latter 128 characters are defined in the Isomorphic ASCII/EBCDIC Conversion Table. (See the MPM write-up of the `ascii_to_ebcDic_` subroutine.) For defined characters, the mappings implemented by `ebcDic_to_ascii_` and `ascii_to_ebcDic_` are isomorphic; i.e., each character has a unique mapping, and mappings are reversible. An undefined (but valid) EBCDIC character is mapped into the ASCII SUB (substitute) character, octal 032; the mapping of such a character is anisomorphic. The result of converting an invalid character is undefined.

```
|-----|  
| ebcdic_to_ascii_ |  
|-----|
```

Usage

```
dcl ebcdic_to_ascii_$table char (256) external static;
```

Note

EBCDIC to ASCII conversion cannot be performed by the PL/1 translate builtin because the EBCDIC string would be scanned under a 7 bit mask. Calling ebcdic\_to\_ascii\_ is extremely efficient, as conversion is performed by a single MVT instruction, and the procedure runs in the stack frame of its caller.

TITLE: Fix privileged initiate bug		STATUS	DATE
AUTHOR: A. Kobziar		Written	05/20/75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR <u>2.2</u> -Fixes Bug Number(s) <u>unreported</u> -Documented in MTB -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input checked="" type="checkbox"/> Better <input type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 05/27/75
		Expires	11/27/75
		DOCUMENTATION CHANGES	
Category (Check One) <input type="checkbox"/> Lib. Maint. Tools <input type="checkbox"/> Sys. Anal. Tools <input type="checkbox"/> Sys. Prog. Tools <input type="checkbox"/> 355 <input type="checkbox"/> BOS <input type="checkbox"/> Salvager <input checked="" type="checkbox"/> Ring Zero <input type="checkbox"/> Ring One <input type="checkbox"/> SysDaemon/Admin. <input type="checkbox"/> Runtime <input type="checkbox"/> User Cmmnd/Subr.		Document	Specify One or More
Objections/Comments:		MPM (Vol, Sect.)	
		PLMS (AN #) 75	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
		Info Segs	
		Other (Name)	
		None (Reason)	Bug Fix

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

\* \* \* \*

SUMMARY:

initiate\$priv\_init\_count's first parameter is declared as a pointer rather than char(\*) for the directory pathname. Change the program to use the correct parameter, as documented.

IMPLICATIONS:

Required for system 2.2

TITLE: Add pi handler to command_query_		STATUS	DATE	
AUTHOR: S. Herbst		Written	05/17/75	
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)		Status	
	Lib. Maint. Tools		Expires	
	Sys. Anal. Tools		DOCUMENTATION CHANGES	
	Sys. Prog. Tools		Document Specify One or More	
	355		MPM (Vol, Sect.) AG93	
BOS		PLMS (AN #)		
Salvager		MOSN (Sect.)		
Ring Zero		MPAM (Sect.)		
Ring One		MSAM (Sect.)		
SysDaemon/Admin.		Info Segs pending changes.info		
Runtime		Other (Name)		
<input checked="" type="checkbox"/> User Cmmd/Subr.		None (Reason)		
Objections/Comments:				

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

\* \* \* \* \*

SUMMARY:

Put a program\_interrupt handler in command\_query\_ that reprints the query and waits for a response.

REASONS:

Queries are sometimes garbled by transmission lines. There is no simple way to reprint the question.



MCR 1183  
Attachment 2

Pending Changes:

A `program_interrupt` handler has been added to `command_query_`. If the user quits before responding to a question and says "pi", the question is repeated.

- (4) status\_code            is the status code which prompted the question; otherwise it should be zero. (Input)
- (5) query\_code            is currently ignored. It is intended for use by specialized handlers for command\_question. (Input)
- 2) answer                is the response (character(\*) varying) read from user\_input. Leading and trailing blanks plus the "new line" character have been removed. (Output)
- 3) caller                is the name (character(\*) of the calling procedure. It may be either varying or nonvarying. (Input)

The remaining arguments are optional as explained in the Notes.

- 4) control\_string        is an ioa\_ control string (character(\*)). See the MPM description of ioa\_. (Input)
- 5) arg<sub>i</sub>                is an ioa\_ format argument. See the MPM description of ioa\_. (Input)

### Notes

The question prepared by command\_query\_ has the format "caller: message". If suppress\_name\_sw is on, then the caller name will be omitted from the question. The message is constructed by ioa\_ from the control string and format arguments. If no control string and therefore no format arguments are given, the message portion of the question is omitted.

If the user issues a quit signal before responding to the question and then invokes the program\_interrupt command, the question is repeated. This feature is useful in case the original question was garbled.

TITLE: Change abbrev to set bit count on profile		STATUS	DATE
AUTHOR: S. Herbst		Written	05/17/75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 05/27/75
		Expires	11/27/75
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason) no interface change	

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

SUMMARY:

Change abbrev to keep an accurate bit count on profiles.

REASONS:

Consistency with the standard meaning of bit count, namely the last meaningful bit in a segment. Currently, the bit count of a user profile is zero.

TITLE: Change name of names\$copy		STATUS	DATE
AUTHOR: S. Herbst		Written	05/20/75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR _____ -Fixes Bug Number(s) _____ -Documented in MTB _____ -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____		Status	A 05/27/75
		Expires	11/27/75
		DOCUMENTATION CHANGES	
Category (Check One)		Document	Specify One or More
<input type="checkbox"/> Lib. Maint. Tools		MPM (Vol, Sect.)	AG92
<input type="checkbox"/> Sys. Anal. Tools		PLMS (AN #)	
<input type="checkbox"/> Sys. Prog. Tools		MOSN (Sect.)	
<input type="checkbox"/> 355		MPAM (Sect.)	
<input type="checkbox"/> BOS		MSAM (Sect.)	
<input type="checkbox"/> Salvager		Info Segs	
<input type="checkbox"/> Ring Zero		Other (Name)	
<input type="checkbox"/> Ring One		None (Reason)	
<input type="checkbox"/> SysDaemon/Admin.			
<input type="checkbox"/> Runtime			
<input checked="" type="checkbox"/> User Cmnd/Subr.			
Objections/Comments:			

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

\* \* \* \*

SUMMARY:

Change the command entry points names\$move and names\$copy to move\_names and copy\_names.

REASONS:

No other commands are called specifying an entry point name.

IMPLICATION:

Incompatible change.

TITLE: Replace calls to hcs\_\$get\_authorization

AUTHOR: Jerry A. Stern

Written 5/20/75  
Status A 05/27/75  
Expires 11/27/75

- Coded in:  PL/I  ALM  other-explain in DETAILED PROPOSAL
- Planned for System MR 2.2
- Fixes Bug Number(s) \_\_\_\_\_
- Documented in MFB \_\_\_\_\_
- User/Operations-visible
- Interface change?  yes  no
- Incompatible change?  yes  no
- Performance:  Better  Same  Worse
- Replaces MCR \_\_\_\_\_

- Category (Check One)
- Lib. Maint. Tools
  - Sys. Anal. Tools
  - Sys. Prog. Tools
  - 355
  - FOS
  - Salvager
  - Ring Zero
  - Ring One
  - SysDaemon/Admin.
  - Runtime
  - User Cmnd/Subr.

DOCUMENTATION CHANGES

- Document Specify One or More
- MFN (Vol, Sect.)
- PLMS (AN #)
- MOSH (Sect.)
- MFAM (Sect.)
- MSAM (Sect.)

Objections/Comments:

- Info Segs
- Other (Name)
- None (Reason)

Use these headings: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (Optional)

SUMMARY: Replace calls to hcs\_\$get\_authorization made by the message segment facility by calls to get\_authorization, get\_max\_authorization, and get\_privileges (defined in MCR 1142).

Reasons: The message segment software currently understands the internal structure of an authorization in order to examine the privilege flags. It is desired to limit this knowledge to the three subroutines mentioned above so that any future changes to the authorization structure will have only a minor impact on programs outside ring 0.

Implications: None

Detailed Proposal: Replace calls to hcs\_\$get\_authorization made by mseg\_add, mseg\_util, and mseg\_create.

TITLE: Install upward compatible object_info_		STATUS	DATE
AUTHOR: M. Weaver		Written	05/20/75
-Coded in: <input checked="" type="checkbox"/> PL/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR 3.0 -Fixes Bug Number(s) -Documented in MTB 187 -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input type="checkbox"/> Same <input checked="" type="checkbox"/> Worse -Replaces MCR		Status	A 05/27/75
		Expires	11/27/75
		DOCUMENTATION CHANGES	
Category (Check One) <input type="checkbox"/> Lib. Maint. Tools <input type="checkbox"/> Sys. Anal. Tools <input type="checkbox"/> Sys. Prog. Tools <input type="checkbox"/> 355 <input type="checkbox"/> BOS <input type="checkbox"/> Salvager <input type="checkbox"/> Ring Zero <input type="checkbox"/> Ring One <input type="checkbox"/> SysDaemon/Admin. <input type="checkbox"/> Runtime <input checked="" type="checkbox"/> User Cmnd/Subr.		Document	Specify One or More
Objections/Comments:		MPM (Vol, Sect.)	SWG
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
		Info Segs	object_segment_changes. info
		Other (Name)	
		None (Reason)	

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

SUMMARY:

Install object\_info\_ changes described in MTB-187, except that the name of the structure has been changed to object\_info from obj\_info and must be declared based in the include file to avoid being allocated. Update the include files object\_info.incl.pll, obj\_map.incl.pll and object\_map.incl.pll. Rename the current include files to object\_info\_old\_1.incl.pll, based\_object\_info\_old\_1.incl.pll, obj\_map\_old\_1.incl.pll and object\_map\_old\_1.incl.pll.

REASONS:

object\_info\_ must reflect new object segment changes, some of which are needed for prelinking. It must be installed for the linker to use so that testing of the translator changes can be completed.

# object\_segment\_changes.info

05/19/75

In a few weeks some of the translators will begin to produce object segments in a slightly different format. These changes will not affect most users, but will affect those who call `object_info_` or who look at entry parameter descriptors. Briefly, the changes are:

- 1) moving the entry parameter descriptor pointers from the definition to the entry sequence,
- 2) optionally having internal static as a separate section,
- 3) adding a gate entry bound indicator to the object map and an `entry_bound` pseudo-op to `alm`, and
- 4) adding the capability for a threaded list of links in the text section (to be snapped only during system initialization).

Change #1 is being made primarily to improve the efficiency of the command processor when it examines entry parameter descriptors. There is a new standard include file, `entry_sequence_info.incl.pl1`, that gives the new declaration of text information associated with the entry sequence. Basically, the parameter descriptor offset array has been moved to the text section, the word before the definition offset points to the descriptor offset array and several new flags have been added.

Change #2 is intended primarily to help the system prelinking project and should not affect most users.

Changes #3 and #4 will not be implemented for a while and will not affect most users.

Changes 2, 3 and 4 affect the object map, which is why the structure returned by `object_info_` must be changed. The include files `obj_map.incl.pl1` and `object_map.incl.pl1` have been updated and the old ones renamed to `obj_map_old_1.incl.pl1` and `object_map_old_1.incl.pl1` for emergency use for translators. The rest of this info file discusses changes in the use of `object_info_`.

To summarize, no items have been deleted from the structure, the `call_delimiter` item has been renamed to `entry_bound` and all of the new items are in the section returned by all three `object_info_` entry points. The include file `object_info.incl.pl1` has been updated, but no one can just recompile. (If one must recompile with the old structure, the old include file has been renamed to `object_info_old_1.incl.pl1`) First, the version number must be filled in by the caller so that `object_info_` will know what version structure to fill in. THIS IS VERY IMPORTANT! The new structure is version 2 and it is recommended that the `version_number` item be set to the variable `object_info_version_2`, which is also declared in the include file. This will make any future version changes very obvious during compilation. For the time being, `object_info_` will treat every version number other than 2 as version 1.

Second, the structure has been renamed to `object_info` and declared based. This is because we want to do away with having two include files for the same structure, one based and one automatic, and to use the `like` attribute instead. There is no way to do this compatibly for everyone; it was thought that it would be better for everyone to change a few lines of declarations than for come to recode their references. The latter may still have to be done if the references aren't completely qualified. Thus most users will change  
`%include object_info;`

```
to  
%include object_info;  
declare 1 of aligned like object_info;
```

The principal content change to the structure is the addition of the static pointer and length, which should be used by everyone interested in an object segment's [original] internal static.



TITLE: Fix bug in BASIC		STATUS	DATE
AUTHOR: M. Weaver		Written	05/21/75
-Coded in: <input type="checkbox"/> PL/I <input checked="" type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR 2.2 -Fixes Bug Number(s) 56 -Documented in MTB -User/Operations-visible Interface change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR		Status	A 05/27/75
		Expires	11/27/75
		DOCUMENTATION CHANGES	
		Document	Specify One or More
		MPM (Vol, Sect.)	
		PLMS (AN #)	
		MOSN (Sect.)	
		MPAM (Sect.)	
		MSAM (Sect.)	
Objections/Comments:		Info Segs	
		Other (Name)	
		None (Reason)	

Use these headings: Summary of Proposal, Reasons for Proposal, Implications, Detailed Proposal.

SUMMARY:

Add fix to bug 056 to version of BASIC authorized by MCR 1146.

REASONS:

There is still time to get latest bug fix into system 2.2.

(056 sin(-abs(x)) = abs(x))

Multics Change Request

TITLE: New Multics Card Input Facility		STATUS	DATE	
AUTHOR: Stanley C. Vestal		Written	05-23-75	
<input checked="" type="checkbox"/> P/L/I <input type="checkbox"/> ALM <input type="checkbox"/> other- explain in DETAILED PROPOSAL -Planned for System MR 2.2 -Fixes Bug Number(s) _____ -Documented in MTB 143 -User/Operations-visible Interface change? <input checked="" type="checkbox"/> yes <input type="checkbox"/> no -Incompatible change? <input type="checkbox"/> yes <input checked="" type="checkbox"/> no -Performance: <input type="checkbox"/> Better <input checked="" type="checkbox"/> Same <input type="checkbox"/> Worse -Replaces MCR _____	Category (Check One)	Status	A 05/27/75	
		Lib. Maint. Tools	Expires	11/27/75
		Sys. Anal. Tools	DOCUMENTATION CHANGES	
		Sys. Prog. Tools	Document	Specify One or More
	355	MPM (Vol, Sect.)	Commands Ref Guide	
	EOS	PLMS (AN #)		
	Salvager	MOSH (Sect.)		
	Ring Zero	MPAM (Sect.)		
	Ring One	MSAM (Sect.)		
	<input checked="" type="checkbox"/> SysDaemon/Admin.	Info Segs	Yes	
	Runtime	Other (Name)		
	User Cmmnd/Subr.	None (Reason)		
Objections/Comments:				

Use these headings: SUMMARY, REASONS, IMPLICATIONS, DETAILED PROPOSAL (Optional)

**SUMMARY:** Modify the Multics Card Input Facility as discussed in MTB 143. Requires extending operator interface in cards overseer and changing the control card logic and storage management techniques in read cards. This will also provide the initial changes to allow bulk card input from the mohawk driver. In the card pool, a user will have a sub-directory per deck access class which will contain the image of the cards read. Access to the directory will be "s" for the user so he will not be able to abuse the system storage. Garbage collection will be based on the age of the segments. A new command "copy\_cards" will be supplied to copy the card image into a user's directory. This implementation will also, fix a bug in raw\_dim to allow reading more than 2457 cards (a full 64k segment).

**Reasons:** Current software places a link to a card image segment in the search path of a user without validation of the privilege to do so. Also, there is currently no provision for placing an access class on a card image segment. These actions are inconsistent with the access controls on Multics.

- Implications:**
1. Causes the deck format for card decks to change.
  2. Requires a new operational procedures for card input.
  3. Requires a new user command for gaining access to a card deck once it has been read into system pool storage.
  4. Requires a new strategy for the management of card image storage to be compatible with the access isolation mechanism.
  5. Card input will no longer be done by IO.SysDaemon. Site should register the user card\_input.Daemon (or such).

Detailed Proposal:

Replace subroutines `cards_overseer_` and `read_cards` in `bound_card_io_` (in `tools`).

`read_cards` will have a new entry point for the mohawk driver to use.

Add command `copy_cards` and subroutine `pool_manager_` to `bound_io_commands_` (in `sss`).

Add subroutine `priv_move_quota_` to `tools` (used by privileged entries of `pool_manager_`).

Command

05/20/75

**Name:** copy\_cards

The copy\_cards command copies specified card image segments from system pool storage into a user's directory. The segments to be copied must have been created using the Multics Card Input Facility. The user process executing this command must have the proper access to the card image segment in order to perform the copy. Multiple copies of the same deck in pool storage will all be copied.

**Usage**

copy\_cards deck\_name [new\_deck\_name]

- 1) deck\_name is the name which was entered on the deck\_id card when the card deck was submitted for reading.
- 2) new\_deck\_name is the pathname of the segment in which the matching card image segment is to be placed. If omitted, the working directory and deck\_name are assumed (optional).

**Notes**

The deck\_name may follow the star convention and all matching card image segments in pool storage to which the user has access will be copied. Similarly, new\_deck\_name may use the equal convention. It is the user's responsibility to resolve name duplication difficulties.

If a deck\_name duplication had occurred during the card reading operation, a message will be printed to inform the user that the contents of deck\_name may be different than intended. Each copy of the duplicate deck may be copied by the user for examination by requesting that deck\_name.\* be copied.

-----  
copy\_cards  
-----

MULTICS PROGRAMMERS' MANUAL

Command  
Page 2  
05/20/75

See the description of the card input facility in the MPM for the format of the control cards needed when submitting a card deck to be read by system operations.

**Example**

```
copy_cards my_deck
```

would copy the user's card image segment named my\_deck from the card pool storage into the user's current working directory.

Subroutine

05/22/75

Name: pool\_manager\_

This subroutine implements an on-line storage management scheme devised for system processes that provide temporary storage for user segments. The principal application of the pool\_manager\_ is the management of card deck image segments read by the system card input process. However, the pool\_manager\_ is sufficiently general to handle other similar applications as well.

A "pool" consists of a subtree of the directory hierarchy. The head directory of this subtree is called the pool root directory. Immediately inferior to the pool root directory are found "access class" directories. Each of these directories has a different access class. Storage is always allocated by pool\_manager\_ within the access class directory that corresponds to the caller's authorization. Below an access class directory are found "person" directories (also called "user pools"). A person directory is created for each person needing temporary storage. A person directory will contain all segments and multi-segment files for a given person at a given access class.

Entry: pool\_manager\_\$init

This entry will ensure that an access class directory exists at the authorization of the caller. If one does not exist, it will be created. This operation requires access to the system\_privilege\_gate whenever the caller authorization does not equal the access class of the pool root directory. The ACL of the access class directory will be set to "sma" for the caller and "s" for \*.\*.\*.

```
declare pool_manager_$init entry (char (*) aligned,  
    fixed bin, bit (36) aligned, fixed bin (35));
```

```
call pool_manager_$init (root, quota, access, code);
```

Where:

Subroutine  
Page 2  
05/22/75

1. root is the pathname of the pool root directory. (Input)
2. quota is the initial value of quota to assign to a newly created access class directory. If the remaining quota on the pool root is less than the value of "quota", the remainder will be moved. (Input)
3. access is the access mode for \*.\* to be placed on person directories in the pool. The first three bits correspond to status, modify and append permission. However, in this case, access is limited to status permission only. (Input)
- 4 code is a standard system error code. (Output)

**Entry:** pool\_manager\_\$open\_user\_pool

This entry will create a person directory beneath the appropriate access class directory if one does not already exist. The ACL of the person directory will be set so that only the caller has modify and append permission. Also, any terminal quota on the person directory will be removed so that the contents of the person directory can be expanded.

**Usage**

```
declare pool_manager_$open_user_pool entry (char (*) aligned,  
char (*), char (*), fixed bin (35));
```

```
call pool_manager_$open_user_pool (root, person, path, code);
```

**Where:**

1. root is as above. (Input)
2. person is the registered person name for whom the pool storage is to be created and managed. (Input)

3. path                    is the pathname of the person pool directory. This is valid only if the value of code is 0. (Output)
4. code                    is as above. (Output)

Entry: pool\_manager\_\$close\_user\_pool

This entry is used to adjust the quota and ACL entries on the person pool directory. If desired, the pool managing process can ensure that pool users are not allowed to create new segments, or cause existing segments to grow.

Usage

```
declare pool_manager_$close_user_pool entry (char (*)
        aligned, char (*), fixed bin, fit (36) aligned,
        fixed bin (35));

call pool_manager_$close_user_pool (root, person, quota,
        access, code);
```

Where:

1. root                    is as above. (Input)
2. person                  is the registered person name of the user pool to be closed. (Input)
3. quota                   is the quota to place on the person directory. If quota = 0, the user will be allowed to grow segments in his pool directory. If quota = 1, the quota on the directory will be set equal to pages used. Any other value will be used as the quota to be set on the directory. (Input)
4. access                  is the access mode for Person.\*.\* on the person directory. The format is as above. Access is limited to status and modify permission only. (Input)



Subroutine  
Page 4  
05/22/75

5. code                    is as above. (Output)

**Entry:** pool\_manager\_\$add\_quota

This entry moves quota to the access class directory corresponding to the calling process. The directory is assumed to be upgraded and hence, access to the system\_privilege\_gate is required. This entry point is intended to be used by record quota overflow condition handlers to restore a useful number of unused pages to the access class pool.

Usage

```
declare pool_manager_$add_quota entry (char (*) aligned,  
                                      fixed bin, fixed bin (35));
```

```
call pool_manager_$add_quota (root, quota, code);
```

Where:

1. root                    is as above. (Input)
2. quota                   is the number of pages to add to the quota of the access class directory. (Input)
3. code                    is as above. (Output)

**Entry:** pool\_manager\_\$clean\_pool

This entry is used to delete items from a specified pool which exceed a specified age limit. Access to the system\_privilege\_gate is required. All person directories having a date-time entry modified older than the age limit are deleted. If a person directory is not deleted, then all contained segment and msf entries are examined. Any such entry having a date-time modified older than the age limit is deleted. All directory and link entries in a person directory are unconditionally deleted. Similarly, all segment and link entries in an access class directory are unconditionally deleted. If an access class directory is left empty as a result of the above deletions, then it too is deleted. Otherwise, its quota is set

to the number of pages used plus a specified grace value.

### Usage

```
declare pool_manager_$clean_pool entry (char (*) aligned,  
      fixed bin, fixed bin, fixed bin (35));
```

```
call pool_manager_$clean_pool (root, age, grace_quota,  
      code);
```

### Where:

1. root                    is as above. (Input)
2. age                    is the maximum age (in days) of segments that  
                          will be permitted to remain in the pool.  
                          (Input)
3. grace\_quota            number of unused pages of quota to remain on  
                          each access class pool directory if not  
                          deleted. (Input)
4. code                    is as above. (Input)

### Entry: pool\_manager\_\$find\_pool

This entry will return the pathname of a person pool directory. The primary purpose of this entry is to remove knowledge of the pool hierarchy structure from user software. No privileged access is required for this entry. A zero error code indicates that the directory exists and is accessible to the user. The value of the error code will be error\_table\_\$moderr or error\_table\_\$no\_info for incorrect access and error\_table\_\$noentry for a missing directory, if the user has status permission to the access class pool. The value of the returned pathname will be correct if the error code is either 0 or error\_table\_\$noentry.

### Usage

```
declare pool_manager_$find_pool entry (char (*) aligned,  
      bit (72) aligned, char (*), char (*),
```

Subroutine  
Page 6  
05/22/75

```
        fixed bin (35));  
    call pool_manager_$find_pool (root, access_class,  
        person, path, code);
```

Where:

1. root                    is as above. (Input)
2. access\_class          is a binary access class which identifies an  
                          access class directory to be searched for the  
                          person directory. (Input)
3. person                 is as above. (Input)
4. path                   is the returned pathname of the person pool  
                          directory. (Output)
5. code                   is as above. (Output)

**Name:** read\_cards

This procedure is used to read cards into Multics segments residing in System Pool Storage. It is called from `cards_overseer_` by the system card reading process and assumes that the card reader is attached. It expects the card decks it encounters to be formatted according to MPM Reference Guide Section 4.4. When called, it continues to read card decks until an error or "buffer\_empty" status is returned from `crz`. It will also return if the first card of a new deck contains "END" in the first three columns.

**Entry:** read\_cards

This entry attempts to read control cards through the `dln_mcc_` and will detach any other stream before reading control cards.

**Usage**

```
declare read_cards entry (char (*) aligned, char (*),  
                          char (*), bit (1) aligned, fixed bin (35));
```

```
call read_cards (root, device_stream, message_stream,  
                test_sw, code);
```

**where:**

1. `root` the pathname of the System Pool Storage root directory. (Input).
2. `device_stream` the name of the stream which is attached to the card input device. (Input)
3. `message_stream` the name of the stream on which error messages will be written. (Input)
4. `test_sw` test switch = "1" if in test mode (currently unused). (Input)

-----  
Subroutine  
Page 2  
05/23/75

5. code                    standard system error return code. (Output)

**Entry:** read\_cards\$callers\_format

This entry is to be used when the card reading process wishes to ensure that the device\_stream is not detached and the control card stream is assumed to be the same as the card deck data stream. In effect, only Multics card code format is supported by this entry and the dim\_name field of the deck\_id is ignored.

**Usage**

```
declare read_cards$callers_format entry (char (*) aligned,  
char (*), char (*), bit (1) aligned,  
fixed bin (35));  
  
call read_cards$callers_format (root, device_stream,  
message_stream, test_sw, code);
```

**where:**

All arguments are as above.

## Procedure

Each deck must begin with two (or more) keypunched control cards: an access\_id card(s) and a deck\_id card. These are used to identify the submitter to MULTICS and to describe the deck name and format. The decks are then submitted to Operations, and will, in general, be read in by the next day. Owing to protection consideration, segments will be created in System Pool Storage rather than placed in the user's directory. No special access need be given to the user's directory for any system process. Once the data has been read, the user may copy the card image segment into his directory with the copy\_cards command (see MPM Command writeup).

Card image segments must be copied from the System Pool Storage within a reasonable time, as the segments in that area will be periodically deleted.

## Control Card Formats

### 1) access\_id card (s)

```
PERSON.PROJECT <ACCESS_CLASS>;
```

where:

PERSON            is the registered person name of the submitter. Only this person will be able to read the card image segment from the pool.

PROJECT           is the project name of the submitter. (optional)

<ACCESS\_CLASS>   is the access class of this data. The ACCESS\_CLASS field may contain imbedded blanks and commas, and may extend over more than one card.

Each access\_id card must end with a comma as the last non-blank character, except for the last card which must end with a semi-colon. Failure to correctly format or specify the access\_id information may result in the inability of the user to access the data from the System Pool Storage.

## 2) deck\_id card

```
DECK_NAME PUNCH_FORMAT
```

where:

**DECK\_NAME** is the entry name to be given to the segment in System Pool Storage. It should be unique among the user's decks recently submitted. In the event of name duplications, the system card reading process will append a numeric component to the end of the supplied name and attempt to create a segment with this new **DECK\_NAME**.

**PUNCH\_FORMAT** is the punch code conversion to use in reading the card deck. It must be one of the following: MCC, VIIPUNCH, RAW.

If name duplications are encountered then there may be more than one deck in System Pool Storage whose first component is **DECK\_NAME**. The `copy_cards` command will retrieve all of these copies when invoked. All characters on the cards will be mapped to lower case except those immediately following an escape character (backslash or cent sign). For example `\MY\_DECK.PL1` will be mapped to `My_Deck.pl1`.

**Example**

Suppose user Doe working on project Proj, wishes to read a FORTRAN source deck into a segment called `alpha.fortran`, with an access class of "proprietary, my\_company".

- 1) The `access_id` card will look like

```
\DOE.\PROJ PROPRIETARY, MY_COMPANY;
```

- 2) The `deck_id` card is

```
ALPHA.FORTRAN MCC
```

where `mcc` is the format of the data cards.

- 3) The control cards followed by the data cards are submitted to operations for reading.

- 4) When the cards have been read, issue the command:

```
copy_cards alpha.fortran
```

on the terminal to copy the deck into the working directory. If the copy does not succeed, then an error message will explain the problem. The user may need to check with operations to correct the problem.

### Notes on Deck Size

Decks must not exceed the maximum length of a Multics segment. A good rule of thumb is to limit decks to single boxes of cards, although more precise counts can be made. For "raw" reading, the actual maximum is 9,792 cards. For Multics card codes, the actual maximum depends on the number of characters actually read, as trailing blanks on cards are ignored. Assuming all 80 columns are punched on each card, the maximum would be 13,055 cards. For 7punched decks, the length of the created segment depends on the length of the original data. The typical 7punch card represents 22 words, but it may represent as many as 4,096 words if the original data contained that many consecutive words of identical contents.

### Errors

The operator will return a note with the deck if any errors took place during the read. In general, the error should be corrected and the deck resubmitted.



**I/O Facilities  
05/09/75**

- 3) Card Reader - Information contained in a deck of cards may be read into the Multics system by submitting the deck to the Multics operations staff. The deck must consist of control cards to identify the the deck, the submitter, and how it is to be read into the system, followed by the cards containing the data punched in one of the Multics standard formats. The content of the deck will be read at some later time into a segment in a special system pool directory. The segment will be given a user-specified name and access class. After the deck has been read, the user may gain access to the data with the `copy_cards` command. The exact format of input decks is described in the MPM Reference Guide Section on Bulk Input and Output.